ADAPTIVE PROCEDURE FOR APPROXIMATING FUNCTIONS BY CONTINUOUS PIECEWISE POLYNOMIALS

by

Qi Chen

and

Ivo Babuška

Technical Note BN-1187

July 1995

Institute for
Physical Science and Technology

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER Technical Note BN-1187	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Adaptive Procedure for Approximating Functions by Continuous Piecewise Polynomials		5. TYPE OF REPORT & PERIOD COVERED Final Life of Contract
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Qi Chen[1] and Ivo Babuska[2]		8. CONTRACT OR GRANT NUMBER(s) [2]ONR/N00014-90-J-1030
9. PERFORMING ORGANIZATION NAME AND ADDRESS [2]Institute for Physical Science and Technology University of Maryland College Park, MD 20742-2431		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Office of Naval Research Arlington, VA 22217		12. REPORT DATE July 1995
		13. NUMBER OF PAGES 14
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT

We consider the problem of approximating function in a general domain in one and two dimensions using piecewise polynomial interpolation. We propose an error estimator and show how to adaptively determine the interpolation degree. Numerical examples are given.

Adaptive Procedure for Approximating Functions by Continuous Piecewise Polynomials

Qi Chen*
Sachs Freeman Associates
1401 McCormick Drive
Landover, MD 20785

Ivo Babuška†
Institute of Physical Science and Technology
University of Maryland
College Park, MD 20742

August 18, 1995

Abstract

We consider the problem of approximating function in a general domain in one and two dimensions using piecewise polynomial interpolation. We propose an error estimator and show how to adaptively determine the interpolation degree. Numerical examples are given.

*Supported by NASA Langley and U.S. Naval Research Laboratory.
†Partially supported by the Office of Naval Research under Grant N00014-90-J1030.

I Introduction

Polynomial interpolation is an important tool in approximating functions. The optimal interpolation in an interval was under much study and was resolved with the proof of the Erdös-Bernstein conjecture [4] [5]. However, few attempts have been made to address the optimal polynomial interpolation in the triangle and in the tetrahedron. In [2] [3], we have computed the positions of the mean optimal interpolation sets in the triangle and in the tetrahedron. The mean optimal sets are close to optimal in the uniform norm and are shown to have the smallest Lebesgue constants among currently known interpolation sets. They perform well in many applications. They have been successfully used in the p-version of the Finite Element Method.

In this paper, we consider the problem of approximating function in a general domain in one and two dimensions using polynomial interpolation. We assume that the domain is partitioned into standard subdomains, *i.e.*, into intervals and triangles. In each subdomain, we approximate the function using the polynomial interpolation points given in [2]. In a partitioned domain, interpolation using the same polynomial degree in every standard subdomain leads to continuous piecewise polynomial. Nevertheless, uniform distribution of degree is usually not economical. In addition, in boundary value problems, small polynomial degree is desired in approximating the essential boundary condition for an efficient implementation of the Finite Element Method. We address the question of how to determine the optimal degree of polynomial interpolation in each subdomain to yield the most efficient approximation.

In section 2, we review the theory of polynomial interpolation and summarize the main results in [2] and [3]. In section 3, we introduce an effective error estimator and present an adaptive procedure for determining the polynomial interpolation degree in each subdomain in R^1 and R^2. We present an algorithm to ensure the continuity of the interpolated piecewise polynomial for a nonuniform distribution of degree.

II On Interpolation

2.1 Interpolation in an interval

Let $I = (-1, 1)$ and $C(\overline{I})$ be the space of continuous functions. Let $C(\overline{I})$ be equipped with the norm $\|f\|_\infty = \max_{t\in\overline{I}} |f(t)|$. Further let $\mathcal{P}_n \subset C(\overline{I})$ be the set of polynomials of degree n. Let $T^n = (\tau_0^n, \tau_1^n, ..., \tau_n^n)$ with $-1 = \tau_0^n < \tau_1^n < ... < \tau_n^n = 1$. Then by $\mathcal{L}_{T^n}$ we denote the mapping $C(\overline{I}) \to \mathcal{P}_n$: $p_n = \mathcal{L}_{T^n} f$ such that $p_n(T^n, f, t) \in \mathcal{P}_n$ and $p_n(T^n, f, \tau_j^n) = f(\tau_j^n), j = 1, ..., n$. Obviously $p_n(T^n, f)$ is uniquely determined and $\mathcal{L}_{T^n}$ is a projection. Denote now

$$\lambda(T) = \|\mathcal{L}_{T^n}\|_\infty = sup_{f\neq 0} \frac{\|\mathcal{L}_T f\|_\infty}{\|f\|_\infty}. \tag{2.1}$$

Let

$$L_k(T^n, t) = \prod_{j=0, j\neq k}^{n} (\frac{t - \tau_j^n}{\tau_k^n - \tau_j^n}), k = 0, ..., n \tag{2.2}$$

be the Lagrange Polynomials associated with the set T^n. It is easy to show that

$$\lambda(T) = \| \sum_{k=0}^{n} |L_k^T(t)| \|_\infty . \tag{2.3}$$

In addition, we introduce

$$\|\langle \mathcal{L}_T \rangle\|_\infty = (\int_{-1}^{1} \sum_{k=0}^{n} |L_k(T,t)|^2 dt)^{\frac{1}{2}}. \tag{2.4}$$

Let $f \in C(I)$ be given and let $\hat{p}_n(f,t) \in \mathcal{P}_n$ be arbitrary, then

$$\|f - \mathcal{L}_T f\|_\infty \leq (1 + \lambda(T^n))\|f - \hat{p}_n\|_\infty . \tag{2.5}$$

(2.5) shows that the interpolation error is up to a constant $(1 + \lambda(T^n))$ the same as the error of the best approximation and hence small $\lambda(T^n)$ is desirable. Further (2.5) also shows that the roundoff error θ (or error of any other kind) in $f(\tau_j^n)$ leads to the increase of the interpolation error at most by $\lambda(T^n)\theta$. This observation will be used in section 3.

Remark. Although (2.5) is only an upper estimate, it can be shown that if $\lambda(T^n)$ rapidly grows as $n \to \infty$, the interpolation can diverge.

Our aim in [2] [3] was to determine the optimal points T^n_{opt} which leads to the *best* interpolation. Of course the term "*best*" has to be defined. For survey of the literature, we refer to [2] [6] [7] [9]. For the purpose of this paper, we say that T^n is optimal if $\lambda(T^n)$ is minimal. More precisely, we denote by T^n_{opt} such that $\lambda_n = \lambda(T^n_{opt}) = \inf \lambda(T^n)$, where *inf* is taken over all interpolations T^n. It can be shown that the set T^n_{opt} exists and its characteristic properties are known as the Erdös-Bernstein conjecture. The conjecture is proved in [4] [5] The points T^n_{opt} and λ_n can be computed numerically. For more details, see [2].

Although we have addressed above only one dimensional case, (2.1) (2.2) (2.5) hold in 2 and 3 dimensions too(with obvious modifications to the definition of the Lagrange polynomials (2.2) and the integral in (2.4)).

Given T_1^n and T_2^n, we say T_1^n is worse than T_2^n if $\lambda(T_1^n) > \lambda(T_2^n)$. T_1^n is close to T_2^n if $\lambda(T_1^n) \approx \lambda(T_2^n)$. This comparison criterion between two sets T^n is useful because $\lambda(T^n)$ can be easily computed. In contrast the optimal set T^n_{opt} is very hard to find especially in 2 and 3 dimensions. No algorithm for locating T^n_{opt} is known in 2 and 3 dimensions. Hence in the literature, various approaches to find approximate optimal sets were proposed and studied (see e.g., [2]). The above criterion gives a characteristic way for selecting the *best known set.*

If we minimize (2.4) instead of (2.3), we get the mean optimal set $T^n_{\langle\mathcal{L}\rangle}$. $T^n_{\langle\mathcal{L}\rangle}$ is much easier to compute numerically. In [2] [3] we have shown that$T^n_{\langle\mathcal{L}\rangle}$ in 2 and 3 dimensions is better than any proposed sets thus far in the literature. In one dimension, T^n_{opt} and λ_n are known, we can compare $\lambda(T^n_{\langle\mathcal{L}\rangle})$ with λ_n or with the Lebesgue constant of any other set. $T^n_{\langle\mathcal{L}\rangle}$ in fact is quite close to T^n_{opt} in the sense that $\lambda(T^n_{\langle\mathcal{L}\rangle})$ is close to λ_n.

Remark. We defined here only the set $T^n_{\langle\mathcal{L}\rangle}$. Other expressions can be used in the minimization procedure to construct optimal sets. For some of the computed optimal sets, see e.g., [2]). However, $T^n_{\langle\mathcal{L}\rangle}$ appears to be the easiest to compute among proposed optimal sets.

Table 2.1: The Lebesgue constant and coordinates of the optimal set and the mean optimal set in the interval. Since both sets are symmetrical, only interior positive coordinates are listed.

n	$\lambda(T^n_{opt})$	$\lambda(T^n_{\langle\mathcal{L}\rangle}) - \lambda(T^n_{opt})$	T^n_{opt}	$T^n_{\langle\mathcal{L}\rangle}$
3	1.42291957	0.03249	0.4177913013559897	0.4306648
4	1.55949021	0.03269	0.6209113046899123	0.6363260
5	1.67221037	0.04662	0.2689070447719729	0.2765187
			0.7341266671891752	0.7485748
6	1.76813458	0.04628	0.4461215299911067	0.4568660
			0.8034402382691066	0.8161267
7	1.85159939	0.05345	0.1992877299056662	0.2040623
			0.5674306027472533	0.5790145
			0.8488719610366557	0.8598070
8	1.92545762	0.05312	0.3477879716116667	0.3551496
			0.6535334790799030	0.6649023
			0.8802308527184540	0.8896327
9	1.99168499	0.05746	0.1585652886576400	0.1618052
			0.4601498259228992	0.4687316
			0.7166138606253078	0.7273222
			0.9027709752917726	0.9108842
10	2.05170576	0.05718	0.2848880010669259	0.2901556
			0.5466676961746040	0.5556701
			0.7640984545671450	0.7739904
			0.9195087517942991	0.9265519
11	2.10658026	0.06007	0.1317518400537555	0.1340857
			0.3862684522940377	0.3927173
			0.6144355426143385	0.6234070
			0.8006822662356081	0.8097370
			0.9322747830229179	0.9384302
12	2.15711897	0.05985	0.2412235692922764	0.2451541
			0.4684175059008267	0.4754842
			0.6683666194633162	0.6770614
			0.8294354799669058	0.8376926
			0.9422316279551781	0.9476477
13	2.20395521	0.06191	0.1127327065284049	0.1144909
			0.3325418228947248	0.3375168
			0.5356654831037281	0.5429843
			0.7119103140476186	0.7202033
			0.8524275899174107	0.8599508
			0.9501460608151026	0.9549426

n	$\lambda(T^n_{opt})$	$\lambda(T^n_{\langle\mathcal{L}\rangle}) - \lambda(T^n_{opt})$	T^n_{opt}	$T^n_{\langle\mathcal{L}\rangle}$
14	2.24759321	0.06173	0.2091510118057353	0.2121872
			0.4091565377641974	0.4147776
			0.5912705457477183	0.5986083
			0.7475281167521386	0.7553639
			0.8710916063656573	0.8779513
			0.9565402633332384	0.9608141
15	2.28844092	0.06328	0.0985298474573020	0.0999008
			0.2918015306737818	0.2957382
			0.4738546882316757	0.4798402
			0.6376896724307452	0.6449010
			0.7770061889653626	0.7843697
			0.8864437409774569	0.8927090
			0.9617797380927199	0.9656095
16	2.32683304	0.06313	0.1845990864374410	0.1870111
			0.3629096640933456	0.3674590
			0.5288572896841651	0.5350106
			0.6767882780854777	0.6837852
			0.8016617897222662	0.8085605
			0.8992200402941425	0.9049549
			0.9661264749901083	0.9695763
17	2.36304752	0.06432	0.0875146934912087	0.0886130
			0.2598842018797722	0.2630690
			0.4243548709184729	0.4293012
			0.5759276542381555	0.5821132
			0.7099951678453442	0.7167274
			0.8224812942273985	0.8289349
			0.9099637674997672	0.9152259
			0.9697722141026608	0.9728948
18	2.39731771	0.06420	0.1652019161293088	0.1671625
			0.3258963986012215	0.3296409
			0.4776989334135101	0.4828825
			0.6164674680899757	0.6225929
			0.7384152664484192	0.7448572
			0.8402138571728484	0.8462483
			0.9190827139401264	0.9239234
			0.9728598818330955	0.9756989
19	2.42984142	0.06515	0.0787200614528085	0.0796194
			0.2342214072823386	0.2368471
			0.3839541516755896	0.3880920
			0.5242304777869164	0.5295337
			0.6515953324320913	0.6575991
			0.7629113849148811	0.7690531
			0.8554359734390852	0.8610795
			0.9268876556810802	0.9313521
			0.9754977704558682	0.9780895

In table 2.1, we give T^n_{opt}, $T^n_{\langle\mathcal{L}\rangle}$ and λ_n, $\lambda(T^n_{\langle\mathcal{L}\rangle})$. Because T^n_{opt}, $T^n_{\langle\mathcal{L}\rangle}$ are symmetrical, we only give the interior positive coordinates, *i.e.*, negative coordinates and points on the boundary ($\tau_0 = -1$ and $\tau_n = 1$) and the center ($\tau_{n/2} = 0$ for even degree) are not listed.

2.2 Interpolation in the triangle

Consider now the standard triangle $S^2 = \{(x, y) : x \geq 0, y \geq 0, 1-x-y \geq 0\}$. $(x, y, 1-x-y)$ are called the barycentric coordinates for the triangle. We denote them as (b_1, b_2, b_3). We seek the set of interpolation points which minimize (2.4) written in the two dimensional form. Analogous to the one dimensional case where we constrain the points τ_0^n and τ_n^n on the boundary of I, we use the points constructed in section 2.1 as the interpolation points on the sides of S^2. We then find the points inside S^2 by minimizing (2.4) properly adjusted to the two dimensional case. We have shown in [2] that there are many local minima. We select the one which leads to the minimal $\lambda(T^n)$ among T^n with various symmetries. We show that these points are the best points known today in the sense defined in section 2.1. We give $T^n_{\langle\mathcal{L}\rangle}$ in table 2.2.

III The adaptive procedure

3.1 The one dimensional case

Let $\Omega = [a, b]$ be partitioned into elements $e_l = [z_1^{(l)}, z_2^{(l)}]$, $l = 1, ..., m$. We assume that the partition has the usual properties, *i.e.*, $z_2^{(l+1)} = z_1^{(l)}$, $z_1^{(1)} = a, z_2^{(m)} = b$. Let $I = (-1, 1)$ be the master element. A linear map ψ_l maps $\overline{I}$ onto e_l.

Let $f \in C(\Omega)$ be a continuous function on Ω, f_l its constraint on e_l and $F_l(\xi)$, $|\xi| < 1$ be the preimage of f_l on I. Using the interpolation points T^{n_l} on I, we construct a polynomial $P_{n_l}(T^{n_l}, F_l, \xi)$ of degree n_l and its image $p_{n_l}(f, t), t \in e_l$. Let $\underline{n} = (n_1, ..., n_m)$, then we denote $p_{\underline{n}}(f, t)$ the piecewise polynomial on Ω such that $p_{\underline{n}} = p_{n_l}(f, t), \forall t \in e_l$. Since the interpolation points contain the end points of the interval, $p_{\underline{n}}$ is continuous.

Let

$$\epsilon_{n_l}(f) = \|f - p_{n_l}\|_{e_l,\infty} \tag{3.1}$$

and

$$\epsilon_{\underline{n}}(f) = \max_{l=1,\dots,m} \epsilon_{n_l}(f) = \|f - p_{\underline{n}}\|_{\Omega,\infty}. \tag{3.2}$$

Given the tolerance ϵ, our aim is to construct $p_{\underline{n}}(t)$ so that $\epsilon_{\underline{n}}(f) \leq \epsilon$. By definition, this is equivalent to have $\epsilon_{n_l}(f) \leq \epsilon$. Hence our aim is to construct adaptively an a posterior error estimator with the polynomial $p_{n_l}(f, t)$ and $P_{n_l}(T^{n_l}, F_l, \xi)$ so that $\epsilon_l(F_l) = \|F_l - P_{n_i}(T^{n_l}, F_l)\|_{I,\infty} \leq \epsilon$. To do that, we need to have an error indicator $\eta(P_{n_i}, F_l)$. For $n_l > 2$, we define:

$$\eta_l^1(P_{n_l}, F_l) = \max_{j=1,\dots,n_l-2} |F_l(\tau_j^{n_l-1}) - P_{n_l}(\tau_j^{n_l-1})|, \tag{3.3}$$

$$\eta_l^2(P_{n_l}, F_l) = \max_{j=1,\dots,s-1;s=2,\dots,n_l-1} |F_l(\tau_j^s) - P_{n_l}(\tau_j^s)|; \tag{3.4}$$

Table 2.2: The Lebesgue constant and barycentric coordinates of the mean optimal set in the triangle. Points with symmetry are listed only once. Other Points are obtained by permuting the barycentric coordinates. n_1, n_3, n_6 are the number of points of singlet, three fold symmetry and six fold symmetry.

n	λ	N_n^2	n_1	n_3	n_6	b_1	b_2	b_3
2	$1\frac{2}{3}$	6	0	0	0	1.0000000	0.0000000	0.0000000
						0.5000000	0.5000000	0.0000000
3	2.1115	10	1	0	0	1.0000000	0.0000000	0.0000000
						0.7251957	0.2748043	0.0000000
			1			0.3333333	0.3333333	0.3333333
4	2.6920	15	0	1	0	1.0000000	0.0000000	0.0000000
						0.8306024	0.1693976	0.0000000
						0.5000000	0.5000000	0.0000000
				1		0.2208880	0.2208880	0.5582239
5	3.3010	21	0	2	0	1.0000000	0.0000000	0.0000000
						0.8866427	0.1133573	0.0000000
						0.6431761	0.3568239	0.0000000
				2		0.1525171	0.1525171	0.6949657
						0.4168658	0.4168658	0.1662683
6	3.7910	28	1	1	1	1.0000000	0.0000000	0.0000000
						0.9194021	0.0805979	0.0000000
						0.7349105	0.2650895	0.0000000
						0.5000000	0.5000000	0.0000000
			1			0.3333333	0.3333333	0.3333333
				1		0.1097139	0.1097139	0.7805723
					1	0.3157892	0.5586077	0.1256031
7	4.3908	36	0	3	1	1.0000000	0.0000000	0.0000000
						0.9398927	0.0601073	0.0000000
						0.7957614	0.2042386	0.0000000
						0.6042138	0.3957862	0.0000000
				3		0.0817370	0.0817370	0.8365261
						0.4494208	0.4494208	0.1011584
						0.2663399	0.2663399	0.4673202
					1	0.2447528	0.6584392	0.0968080
8	5.0893	45	0	3	2	1.0000000	0.0000000	0.0000000
						0.9533797	0.0466203	0.0000000
						0.8375919	0.1624081	0.0000000
						0.6801403	0.3198597	0.0000000
						0.5000000	0.5000000	0.0000000
				3		0.0627331	0.0627331	0.8745338
						0.2153606	0.2153606	0.5692789
						0.3891297	0.3891297	0.2217406
					2	0.3657423	0.5524728	0.0817849
						0.1942206	0.7294168	0.0763626

n	λ	N_n^2	n_1	n_3	n_6	b_1	b_2	b_3
9	5.9181	55	1	3	3	1.0000000	0.0000000	0.0000000
						0.9626819	0.0373181	0.0000000
						0.8672666	0.1327334	0.0000000
						0.7361751	0.2638249	0.0000000
						0.5815151	0.4184849	0.0000000
			1			0.3333333	0.3333333	0.3333333
				3		0.0493729	0.0493729	0.9012542
						0.4658361	0.4658361	0.0683277
						0.1769439	0.1769439	0.6461122
					3	0.3020146	0.6309227	0.0670627
						0.1575680	0.7808733	0.0615587
						0.3261032	0.4887991	0.1850977
10	7.0851	66	0	4	4	1.0000000	0.0000000	0.0000000
						0.9693919	0.0306081	0.0000000
						0.8889846	0.1110154	0.0000000
						0.7782484	0.2217516	0.0000000
						0.6451372	0.3548628	0.0000000
						0.5000000	0.5000000	0.0000000
				4		0.0397231	0.0397231	0.9205538
						0.1477532	0.1477532	0.7044935
						0.4210577	0.4210577	0.1578846
						0.2859582	0.2859582	0.4280837
					4	0.3962235	0.5463689	0.0574076
						0.2531675	0.6909248	0.0559077
						0.1304041	0.8190269	0.0505691
						0.2760598	0.5678554	0.1560848
11	8.3383	78	0	5	5	1.0000000	0.0000000	0.0000000
						0.9743976	0.0256024	0.0000000
						0.9054668	0.0945332	0.0000000
						0.8104474	0.1895526	0.0000000
						0.6950282	0.3049718	0.0000000
						0.5665299	0.4334701	0.0000000
				5		0.0325950	0.0325950	0.9348099
						0.4754886	0.4754886	0.0490228
						0.1252588	0.1252588	0.7494824
						0.2469949	0.2469949	0.5060102
						0.3752681	0.3752681	0.2494638
					5	0.3404173	0.6107764	0.0488062
						0.2152428	0.7374393	0.0473178
						0.1097836	0.8480326	0.0421838
						0.3649733	0.4997724	0.1352543
						0.2363509	0.6303341	0.1333150

n	λ	N_n^2	n_1	n_3	n_6	b_1	b_2	b_3
12	10.082	91	1	4	7	1.0000000	0.0000000	0.0000000
						0.9782397	0.0217603	0.0000000
						0.9184891	0.0815109	0.0000000
						0.8356932	0.1643068	0.0000000
						0.7347400	0.2652600	0.0000000
						0.6208376	0.3791624	0.0000000
						0.5000000	0.5000000	0.0000000
			1			0.3333333	0.3333333	0.3333333
				4		0.0271978	0.0271978	0.9456044
						0.1075744	0.1075744	0.7848512
						0.4415257	0.4415257	0.1169486
						0.2152525	0.2152525	0.5694951
					7	0.4166350	0.5411712	0.0421938
						0.2954879	0.6624810	0.0420312
						0.1853001	0.7741774	0.0405225
						0.0937098	0.8706643	0.0356259
						0.3187835	0.5641899	0.1170266
						0.2045479	0.6802371	0.1152150
						0.3305135	0.4512984	0.2181880
13	12.046	105	0	6	8	1.0000000	0.0000000	0.0000000
						0.9812954	0.0187046	0.0000000
						0.9289266	0.0710734	0.0000000
						0.8560408	0.1439592	0.0000000
						0.7665724	0.2334276	0.0000000
						0.6649507	0.3350493	0.0000000
						0.5559156	0.4440844	0.0000000
				6		0.0230602	0.0230602	0.9538797
						0.4816638	0.4816638	0.0366724
						0.0934032	0.0934032	0.8131936
						0.1893266	0.1893266	0.6213469
						0.4039822	0.4039822	0.1920355
						0.2969227	0.2969227	0.4061545
					8	0.3679120	0.5953680	0.0367200
						0.2590310	0.7043884	0.0365805
						0.1611020	0.8038486	0.0350494
						0.0809091	0.8887075	0.0303834
						0.3920816	0.5059948	0.1019235
						0.2807129	0.6169627	0.1023245
						0.1787576	0.7205294	0.1007129
						0.2928111	0.5148749	0.1923141

For $n_l = 2$,

$$\eta_l^1(P_{n_l=2}, F_l) = \eta_l^2(P_{n_l=2}, F_l) = \max_{j=1,2} |F_l(\tau_j^3) - P_{n_l}(\tau_j^3)| \tag{3.5}$$

Obviously, $\eta_1^l \leq \eta_2^l \leq \|F_l - P_{n_l}(T^{n_l}, F_l)\|_{I,\infty} = \epsilon_l$. We recommend using η_2^l since it is much more effective than η_1^l especially for high degree interpolation. The idea behind the proposed form of the error indicator is the following: If the interpolation error in element e_l is too large, we increase the degree of the polynomial. Since in most cases, getting $F_l(\tau)$ is expensive(*e.g.*, in solid modeling, F_l is obtained using interrogation operators of the solid modeler),we use only the previously computed values $F_l(\tau_j^{n_l-1})$ in the adaptive process. For degree 2, (3.3) and (3.4) are not defined, we use degree 3 points as in the error indicator. This increases a little computation time. Since the interpolation degree 3 is low, this is not a serious impediment. By this procedure, which is parallel, we construct the adaptive interpolation with n_l depending on the given tolerance and the function f.

Note the optimal interpolation points used for interpolation satisfy $-1 < \tau_1^{n_l} < \tau_1^{n_l-1} < \tau_2^{n_l} < \tau_2^{n_l-1} < ... < \tau_{n_l-2}^{n_l-1} < \tau_{n_l-1}^{n_l} < 1$. Therefore, the error indicator η^1 and η^2 never sample points in intervals (-1,$\tau_1^{n_l}$) and $(\tau_{n_l-1}^{n_l}, 1)$. This can remedied by introducing new estimators

$$\tilde{\eta}_l^1(P_{n_l}, F_l) = \max_{j=1,\dots,n_l} |F_l(\tau_j^{n_l+1}) - P_{n_l}(\tau_j^{n_l+1})|, \tag{3.6}$$

$$\tilde{\eta}_l^2(P_{n_l}, F_l) = \max(\tilde{\eta}_l^1(P_{n_l}, F_l), \eta_l^2(P_{n_l}, F_l)). \tag{3.7}$$

However, Since interpolation points are denser near the end points than near the center, the intervals (-1,$\tau_1^{n_l}$) and ($\tau_{n_l-1}^{n_l}$,1) are quite small (compared to the average distance between neighboring interpolation points). In most cases, the results of using $\eta_l^2(P_{n_l}, F_l)$ and $\tilde{\eta}_l^2(P_{n_l}, F_l)$ are quite similar. However, $\tilde{\eta}_l^2(P_{n_l}, F_l)$ has the disadvantage of using higher degree information not computed previously in the adaptive process. When getting $F_l(\tau)$ is not expensive, it may be advantagious to use $\tilde{\eta}_l^2(P_{n_l}, F_l)$

Example 3.1. Let $\Omega = [0,8]$, $l_1 = (0,2), l_2 = (2,4), l_3 = (4,6), l_4 = (6,8)$. Let $f = \frac{1}{(x-10)^2+1}$. In table 3.1, we report the values $\epsilon_l, \eta_l^1, \eta_l^2$ as a function of the polynomial degree n,

We see that both error indicators are quite reliable. The effective indices(the ratio of the error indicator η_l and the actual error ϵ_l) are near one. The second error indicator η_l^2 is more effective than η_l^1. Although the effective indices are not far from one, we suspect they are not asymptotically exact, i.e., it does not approach to one as interpolation degree increases to infinity.

We also see that uniform degree interpolation is not economical. In table 3.2, we give the optimal degree distribution for various tolerance ϵ using η_l^2 as the error estimator.

Example 3.1 is typical and similar results are obtained for other test cases.

3.2 The two dimensional case

Let $\Omega \subset R^2$ be a closed polygonal domain partitioned into triangular elements e_l in the standard way. Let $E_l^k, k = 1,2,3$ be the edges of e_l and let ϵ be the tolerance. Further let D be the standard triangle and ψ_l maps D onto e_l. As before, let $f \in C(\Omega)$ be a continuous function on Ω, f_l its restriction on e_l and F_l its preimage on D. In exactly the same way as

Table 3.1: Errors and error indicators for example 3.1.

n	ϵ_1	η_1^1	η_1^1/ϵ_1	η_1^2	η_1^2/ϵ_1	ϵ_2	η_2^1	η_2^1/ϵ_2	η_2^2	η_2^2/ϵ_2
2	0.27E-04	0.24E-04	0.87	0.24E-04	0.87	0.95E-04	0.82E-04	0.87	0.82E-04	0.87
3	0.17E-05	0.15E-05	0.92	0.15E-05	0.92	0.74E-05	0.66E-05	0.90	0.66E-05	0.90
4	0.10E-06	0.88E-07	0.86	0.88E-07	0.86	0.57E-06	0.48E-06	0.85	0.48E-06	0.85
5	0.61E-08	0.56E-08	0.91	0.56E-08	0.91	0.43E-07	0.38E-07	0.89	0.38E-07	0.89
6	0.36E-09	0.33E-09	0.91	0.35E-09	0.97	0.32E-08	0.29E-08	0.89	0.31E-08	0.96
7	0.21E-10	0.19E-10	0.91	0.21E-10	0.98	0.24E-09	0.21E-09	0.88	0.23E-09	0.97
8	0.12E-11	0.11E-11	0.92	0.12E-11	0.94	0.17E-10	0.15E-10	0.90	0.16E-10	0.94
9	0.71E-13	0.64E-13	0.91	0.68E-13	0.96	0.12E-11	0.11E-11	0.89	0.11E-11	0.96
n	ϵ_3	η_3^1	η_3^1/ϵ_3	η_3^2	η_3^2/ϵ_3	ϵ_4	η_4^1	η_4^1/ϵ_4	η_4^2	η_4^2/ϵ_4
2	0.49E-03	0.42E-03	0.86	0.42E-03	0.86	0.50E-02	0.42E-02	0.84	0.42E-02	0.84
3	0.52E-04	0.45E-04	0.86	0.45E-04	0.86	0.75E-03	0.60E-03	0.80	0.60E-03	0.80
4	0.54E-05	0.44E-05	0.83	0.44E-05	0.83	0.10E-03	0.82E-04	0.81	0.82E-04	0.81
5	0.54E-06	0.45E-06	0.84	0.45E-06	0.84	0.11E-04	0.99E-05	0.87	0.99E-05	0.87
6	0.52E-07	0.45E-07	0.86	0.49E-07	0.94	0.97E-06	0.93E-06	0.96	0.96E-06	0.99
7	0.49E-08	0.41E-08	0.85	0.47E-08	0.97	0.48E-07	0.41E-07	0.84	0.48E-07	0.99
8	0.43E-09	0.38E-09	0.88	0.41E-09	0.93	0.26E-07	0.15E-07	0.60	0.22E-07	0.84
9	0.37E-10	0.32E-10	0.87	0.36E-10	0.96	0.75E-08	0.51E-08	0.68	0.67E-08	0.89

Table 3.2: Adaptive interpolation degrees and errors for various tolerances for example 3.1.

ϵ	$\epsilon_{\underline{n}}$	n_1	ϵ_1	n_2	ϵ_2	n_3	ϵ_3	n_4	ϵ_4
1.0E-3	7.50E-4	2	2.71E-5	2	9.46E-5	2	4.92E-4	3	7.50E-4
1.0E-4	1.00E-4	2	2.71E-5	2	9.46E-5	3	5.22E-5	3	1.00E-4
1.0E-5	1.15E-5	3	1.66E-6	3	7.40E-6	4	5.38E-6	5	1.15E-5
1.0E-6	9.74E-7	4	1.02E-7	4	5.71E-7	5	5.37E-7	6	9.74E-7
1.0E-7	1.02E-7	4	1.02E-7	5	4.33E-8	6	5.22E-8	7	4.84E-8
1.0E-8	7.59E-9	5	6.12E-9	6	3.23E-9	7	4.88E-9	9	7.59E-9

in the one dimensional case, from the error indicator $\eta^{(l)}$ ($\eta_1^{(l)}$,$\eta_2^{(l)}$) defined analogous to (3.4), we construct a polynomial $P_{n_l}(T^{n_l}, F_l)$ such that $\epsilon(F_l) = \|F_l - P_{n_l}(T^{n_l}, F_l)\|_{e_l,\infty} \approx \eta^{(l)} \leq \epsilon$. By p_{n_l}, we denote the image of P_{n_l} on e_l and by $p_{\underline{n}}(f,t)$, we denote the piecewise polynomial function on Ω such that its restriction on e_l is p_{n_l}.

In contrast to the one dimensional case, if the polynomial degrees for two elements sharing a common edge are different, $p_{\underline{n}}(f,t)$ is no longer continuous on the common edge. We need to modify p_{n_l} in the adaptive procedure to construct a new piecewise polynomial $\overline{p}_{\underline{n}}$ so that on the common edge of the two elements $\overline{p}_{\underline{n}}$ is continuous.

Note that during the adaptive process, the interpolation degree n_l for each element is given (starting from, say, $n_l = 2$ for all elements). We observe that on the common edge $E = E_{l_1}^{k_1} = E_{l_2}^{k_2}$, both $P_{n_{l_1}}^E$,$P_{n_{l_2}}^E$ are within tolerance ϵ of the function F. Therefore we use polynomial $P_{n_E}^E$ where $n_E = \min(n_{l_1}, n_{l_2})$ to approximate F on the common edge E. After interpolating the function with degree n_E on every edge, we interpolate the function in each element e_l by the following procedure. For a node on edge E, we replace the function value F at that node with the value of the edge interpolated function $P_{n_E}^E$. For a node in the interior, we use the original function value F. By this procedure, which is parallel, we obtain a continuous polynomial $\overline{p}_{\underline{n}}$ of degree n_l on e_l. We have $\|f - \overline{p}_{\underline{n}}\|_{\Omega,\infty} \leq \epsilon(1 + \lambda(T^{\overline{n}}))$ where $\overline{n} = \max n_e$. The error $\|f - \overline{p}_{n_l}\|_{e_j,n}$ can be estimated by the error indicator. If $\|f - \overline{p}_{n_l}\|_{e_l,\infty} \leq \epsilon$ is not satisfied for some element e_l, we increase the approximation degree n_l and continue the adaptive procedure.

Remark. $\|f - \overline{p}_{\underline{n}}\|_{\Omega,\infty} \leq \epsilon(1 + \lambda(T^{\overline{n}}))$ is an over estimate. Actually, let $\epsilon_1 = \max \epsilon_E$, where ϵ_E is the error on the edge E, $\epsilon_1 \leq \epsilon$. then $\|f - \overline{p}_{\underline{n}}\|_{\Omega,\infty} \leq \epsilon + \lambda(T^{\overline{n}})\epsilon_1$. ϵ_1 is usually much samller than ϵ because the Lebesgue function on the triangle edges is usually much smaller than the Lebesgue constant. The bound can further be made sharper. Therefore, $\|f - p_{\underline{n}}\|_{\Omega,\infty} \leq \epsilon$ is more likely to be satisfied.

Example 3.2. Let $\Omega = [0,4] \times [0,4]$. Ω is partitioned into 8 triangles $e_1 = \{(x,y) : x \geq 0, y \geq 0, x + y \leq 2\}$, $e2 = \{(x,y) : x \leq 2, y \leq 0, x + y \geq 2\}$, $e_3 = e_1 + (2,0)$, $e_4 = e_2 + (2,0)$, $e_5 = e_1 + (0,2)$, $e_6 = e_2 + (0,2)$, $e_7 = e_1 + (2,2)$, $e_8 = e_2 + (2,2)$. Let $f = \frac{1}{((x+1)^2+1)((y+1)^2+1)}$. In table 3.3 we show the error and error indicators for $f - p_{n_l}$. For various tolerance ϵ, the sequence of the adaptive approximation degree is given in table 3.4. We also report the error and the indicators for the adaptively determined $\overline{p}_{\underline{n}}$ using η_l^2.

Note the error indicators in the triangle are usually not as effective as in the one dimensional case.

Remark. Interpolation in domains partitioned into curvilinear elements is done in the same way as in the finite element method using pullback polynomial on the standard element.

Remark. Often we have to impose an upper bound on the degree of used polynomials. If then the accuracy is not achieved, the mesh has to be refined in those elements where the desired accuracy is not achieved.

Table 3.3: Errors and error indicators for example 3.2.

n	ϵ_1	η_1^1	η_1^1/ϵ_1	η_1^2	η_1^2/ϵ_1	ϵ_2	η_2^2	η_2^1/ϵ_2	η_2^2	η_2^2/ϵ_2
2	0.97E-02	0.82E-02	0.84	0.82E-02	0.84	0.19E-02	0.16E-02	0.88	0.16E-02	0.88
3	0.12E-02	0.12E-02	1.00	0.12E-02	1.00	0.27E-03	0.24E-03	0.90	0.24E-03	0.90
4	0.25E-03	0.13E-03	0.52	0.13E-03	0.52	0.33E-04	0.24E-04	0.74	0.24E-04	0.74
5	0.13E-03	0.53E-04	0.40	0.53E-04	0.40	0.19E-04	0.18E-04	0.95	0.18E-04	0.95
6	0.41E-04	0.21E-04	0.51	0.25E-04	0.62	0.77E-05	0.52E-05	0.68	0.74E-05	0.96
7	0.96E-05	0.55E-05	0.58	0.81E-05	0.84	0.23E-05	0.16E-05	0.69	0.20E-05	0.89
8	0.20E-05	0.12E-05	0.58	0.18E-05	0.87	0.52E-06	0.34E-06	0.65	0.45E-06	0.85
9	0.27E-06	0.17E-06	0.64	0.26E-06	0.94	0.85E-07	0.49E-07	0.57	0.78E-07	0.92
n	ϵ_3	η_3^1	η_3^1/ϵ_3	η_3^2	η_3^2/ϵ_3	ϵ_4	η_4^2	η_4^1/ϵ_4	η_4^2	η_4^2/ϵ_4
2	0.18E-02	0.16E-02	0.90	0.16E-02	0.90	0.70E-03	0.63E-03	0.90	0.63E-03	0.90
3	0.24E-03	0.24E-03	1.00	0.24E-03	1.00	0.92E-04	0.92E-04	1.00	0.92E-04	1.00
4	0.31E-04	0.12E-04	0.39	0.12E-04	0.39	0.12E-04	0.85E-05	0.72	0.85E-05	0.72
5	0.16E-04	0.10E-04	0.62	0.10E-04	0.62	0.50E-05	0.39E-05	0.79	0.39E-05	0.79
6	0.57E-05	0.42E-05	0.73	0.51E-05	0.88	0.20E-05	0.16E-05	0.83	0.19E-05	0.99
7	0.15E-05	0.11E-05	0.72	0.13E-05	0.81	0.65E-06	0.42E-06	0.65	0.49E-06	0.75
8	0.34E-06	0.23E-06	0.70	0.23E-06	0.70	0.17E-06	0.90E-07	0.54	0.12E-06	0.69
9	0.46E-07	0.35E-07	0.75	0.36E-07	0.79	0.28E-07	0.13E-07	0.47	0.20E-07	0.71
n	ϵ_5	η_5^1	η_5^1/ϵ_5	η_5^2	η_5^2/ϵ_5	ϵ_6	η_6^2	η_6^1/ϵ_6	η_6^2	η_6^2/ϵ_6
2	0.18E-02	0.16E-02	0.90	0.16E-02	0.90	0.70E-03	0.63E-03	0.90	0.63E-03	0.90
3	0.24E-03	0.24E-03	1.00	0.24E-03	1.00	0.92E-04	0.92E-04	1.00	0.92E-04	1.00
4	0.31E-04	0.12E-04	0.39	0.12E-04	0.39	0.12E-04	0.85E-05	0.72	0.85E-05	0.72
5	0.16E-04	0.10E-04	0.62	0.10E-04	0.62	0.50E-05	0.39E-05	0.79	0.39E-05	0.79
6	0.57E-05	0.42E-05	0.73	0.51E-05	0.88	0.20E-05	0.16E-05	0.83	0.19E-05	0.99
7	0.15E-05	0.11E-05	0.72	0.13E-05	0.81	0.65E-06	0.42E-06	0.65	0.49E-06	0.75
8	0.34E-06	0.23E-06	0.70	0.23E-06	0.70	0.17E-06	0.90E-07	0.54	0.12E-06	0.69
9	0.46E-07	0.35E-07	0.75	0.36E-07	0.79	0.28E-07	0.13E-07	0.47	0.20E-07	0.71
n	ϵ_7	η_7^1	η_7^1/ϵ_7	η_7^2	η_7^2/ϵ_7	ϵ_8	η_8^2	η_8^1/ϵ_8	η_8^2	η_8^2/ϵ_8
2	0.18E-03	0.13E-03	0.71	0.13E-03	0.71	0.58E-04	0.49E-04	0.84	0.49E-04	0.84
3	0.24E-04	0.17E-04	0.72	0.17E-04	0.72	0.78E-05	0.66E-05	0.85	0.66E-05	0.85
4	0.30E-05	0.21E-05	0.68	0.21E-05	0.68	0.99E-06	0.79E-06	0.80	0.79E-06	0.80
5	0.36E-06	0.26E-06	0.72	0.26E-06	0.72	0.14E-06	0.10E-06	0.72	0.11E-06	0.75
6	0.39E-07	0.29E-07	0.73	0.30E-07	0.75	0.18E-07	0.11E-07	0.62	0.16E-07	0.91
7	0.45E-08	0.29E-08	0.65	0.37E-08	0.82	0.21E-08	0.12E-08	0.56	0.19E-08	0.89
8	0.45E-09	0.27E-09	0.59	0.36E-09	0.81	0.21E-09	0.12E-09	0.56	0.18E-09	0.84
9	0.33E-10	0.19E-10	0.59	0.30E-10	0.91	0.17E-10	0.87E-11	0.52	0.15E-10	0.92

Table 3.4: Adaptive interpolation degrees and errors for various tolerances for example 3.2. ϵ is the given tolerance. $\epsilon_{\underline{n}}$ is the error in Ω.

ϵ	n_1	ϵ_1	n_2	ϵ_2	n_3	ϵ_3	n_4	ϵ_4
1.0E-3	4	0.25E-03	3	0.26E-03	3	0.24E-03	2	0.70E-03
1.0E-4	5	0.13E-03	4	0.41E-04	4	0.31E-04	3	0.92E-04
1.0E-5	7	0.96E-05	6	0.78E-05	6	0.57E-05	4	1.08E-05
1.0E-6	9	0.27E-06	8	0.52E-06	8	0.34E-06	7	0.66E-06
$\epsilon_{\underline{n}}$	ϵ_1	n_2	ϵ_2	n_3	ϵ_3	n_4	ϵ_4	
0.70E-3	3	0.24E-03	2	0.70E-03	2	0.18E-03	2	0.58E-04
0.92E-4	4	0.31E-04	3	0.92E-04	3	0.24E-04	2	0.59E-04
1.08E-5	6	0.57E-05	4	1.08E-05	4	0.30E-05	3	0.78E-05
0.66E-6	8	0.34E-06	7	0.66E-06	5	0.36E-06	4	1.01E-06

References

[1] Angelos, J.R., Kaufman, E.H. Jr., Henry, M.S. and Lenker, T.D. (1989), Optimal Nodes for Polynomial Interpolation. Approximation Theory VI, ed. Chui, Schumaker, and Ward, Academix Press, pp17-20.

[2] Chen, Q. and Babuška, I. (1995), Approximate Optimal Points for Polynomial Intepolation of Real Functions in the Interval and in the Triangle.

[3] Chen, Q. and Babuška, I. (1995), The Optimal Symmetrical Points for Polynomial Intepolation of Real Functions in the Tetrahedron.

[4] De Boor, C. and Pinkus, A. (1978), Proof of Bernstein and Erdös Concerning the Optimal Nodes for Polynomial Interpolation. J. Approx. Theo. 24, 289-303.

[5] Kilgore, T.A. (1978), A Characterization of the Lagrange Interpolating Projection with Minimal Chebycheff Norm. J. Approx. Theo. 24, 273-288.

[6] Sendov, Bl. and Andreev, A. (1994), Approximation and Interpolation Theory. in: Ciarlet, P.G. and Lions, J.L., eds., Handbook of Numerical Analysis, Vol. III, North-Holland, Amsterdam, 223-462.

[7] Szabados, J. and Vertesi, P. (1990), Interpolation of Functions. World Scientific, Singapore.

[8] Szabó, B.A. and Babuška, I. (1991), Finite Element Analysis, Wiley, New York.

[9] Tikhomirov, V.M. (1990), Approximation Theory, in Analysis II, Gamkrelidze, R.V. Ed., Encyclopaedia of Mathematical Sciences, vol 14. Springer-Verlag, Berlin.

The Laboratory for Numerical Analysis is an integral part of the Institute for Physical Science and Technology of the University of Maryland, under the general administration of the Director, Institute for Physical Science and Technology. It has the following goals:

> To conduct research in the mathematical theory and computational implementation of numerical analysis and related topics, with emphasis on the numerical treatment of linear and nonlinear differential equations and problems in linear and nonlinear algebra.
>
> To help bridge gaps between computational directions in engineering, physics, etc., and those in the mathematical community.
>
> To provide a limited consulting service in all areas of numerical mathematics to the University as a whole, and also to government agencies and industries in the State of Maryland and the Washington Metropolitan area.
>
> To assist with the education of numerical analysts, especially at the postdoctoral level, in conjunction with the Interdisciplinary Applied Mathematics Program and the programs of the Mathematics and Computer Science Departments. This includes active collaboration with government agencies such as the National Institute of Standards and Technology.
>
> To be an international center of study and research for foreign students in numerical mathematics who are supported by foreign governments or exchange agencies (Fulbright, etc.).